Hepatoprotective effects of systemic ER activation

ChIPseq/Epigenome genome - Annotation of Peak files and feature examination

Christian Sommerauer & Carlos Gallardo

08 November, 2022

Load the enhancers and promoters which were demarcated using various histone modifications. Then annotate these features and export them as tables. For promoters, only one peak per gene, the closest one, is permitted, multiple TSS are disregaded.

```
library("ChIPpeakAnno")
library("GenomicRanges")
options(connectionObserver = NULL) #That is a work-around, as the org.Mm. package cannot be loaded
library("org.Mm.eg.db")
library("biomaRt")
library("tidyverse")

import_list <- list(
promoters_genomewide <- read.delim("results/Epigenome_analysis/Final_promoter_files/prom.3.genomewide.F
enhancers_genomewide <- read.delim("results/Epigenome_analysis/Final_enhancer_files/enh.5.FINAL.genomew
enhancers_allDB <- read.delim("results/Epigenome_analysis/Final_enhancer_files/enh.5.FINAL.H3K27ac_broa
promoters_allDB <- read.delim("results/Epigenome_analysis/Final_promoter_files/prom.8.FINAL.H3K27ac_bro
enhancers_HFDup <- read.delim("results/Epigenome_analysis/Final_enhancer_files/enh.7.HFDup.FINAL.H3K27a
promoters_HFDup <- read.delim("results/Epigenome_analysis/Final_promoter_files/prom.16.FINAL.HFDup.H3K2
enhancers_HFDdown <- read.delim("results/Epigenome_analysis/Final_enhancer_files/enh.9.HFDdown.FINAL.H3
promoters_HFDdown <- read.delim("results/Epigenome_analysis/Final_promoter_files/prom.24.FINAL.HFDdown.

names(import_list) <- c("promoters_genomewide", "enhancers_genomewide", "enhancers_allDB", "promoters_a

##Filter out all locations which are NOT on a chromosome.
import_list2 <- list()
for (i in names(import_list)) {
object <- import_list[[i]] # Save the respective dfs in a non-list object, and put that one into a list
object <- dplyr::filter(object, grepl("chr", V1))
import_list2[[i]] <- object
}


#listEnsemblArchives()
mart <- useMart(biomart = "ensembl", dataset = "mmusculus_gene_ensembl", host = "https://sep2019.archive
# use the getAnnotation function to obtain the TSS
annoData <- getAnnotation(mart, featureType = "TSS")
```

```
# Annotate the peak files.
peaks_annotated <- list()
for (i in names(import_list2)) {
object <- import_list2[[i]]
colnames(object) <- c("chrom", "start", "end")
nrow(object)
object2 <- makeGRangesFromDataFrame(object,  start.field = "start", end.field = "end", ignore.strand = T


#Give ranges numeric names in order
names(object2) <- c(1:length(object2))

#Annotate granges with the nearest TSS
object3 <- annotatePeakInBatch(object2,
                               AnnotationData=annoData,
                               featureType = "TSS",
                               output="nearestLocation",
                               PeakLocForDistance = "start")

object3 <- as.data.frame(object3)
peaks_annotated[[i]] <- object3
}

# For the promoters, remove the duplicated gene names and only allow the closest peak to a given gene.

library(data.table)

peaks_annotated[["promoters_genomewide"]] <- peaks_annotated[["promoters_genomewide"]] %>% dplyr::group_
peaks_annotated[["promoters_genomewide"]] <- setDT(peaks_annotated[["promoters_genomewide"]])[order(shor

peaks_annotated[["promoters_HFDdown"]] <- peaks_annotated[["promoters_HFDdown"]] %>% dplyr::group_by(fea
peaks_annotated[["promoters_HFDdown"]] <- setDT(peaks_annotated[["promoters_HFDdown"]])[order(shortestDi

peaks_annotated[["promoters_HFDup"]] <- peaks_annotated[["promoters_HFDup"]] %>% dplyr::group_by(feature
peaks_annotated[["promoters_HFDup"]] <- setDT(peaks_annotated[["promoters_HFDup"]])[order(shortestDistan

peaks_annotated[["promoters_allDB"]] <- peaks_annotated[["promoters_allDB"]] %>% dplyr::group_by(feature
peaks_annotated[["promoters_allDB"]] <- setDT(peaks_annotated[["promoters_allDB"]])[order(shortestDistan

saveRDS(peaks_annotated, "results/Epigenome_analysis/annotated_diffbind_and_genomewide_promoters_enhanc
```

Export the bedfiles, which are required for further analyses

```
peaks_annotated_bed <- list()
for (i in names(peaks_annotated)) {

  peaks_annotated_bed[[i]] <- peaks_annotated[[i]] %>% dplyr::select("chrom"=seqnames, start, end)

write.table(peaks_annotated_bed[[i]], paste0("results/Epigenome_analysis/",i, "_", as.character(length(u
```

```
}
```

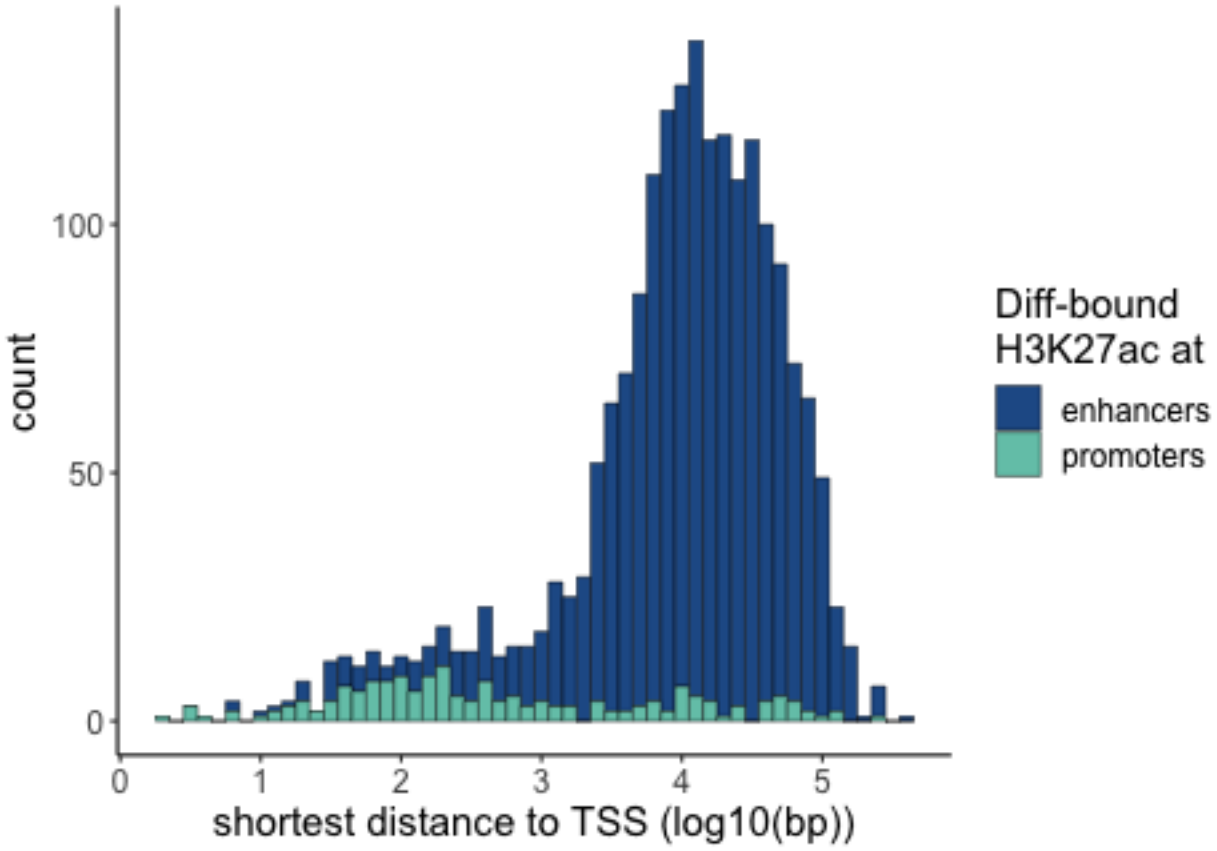
Summary statistics about distance to nearest TSS to evaluate how well the enhancers/promoters were determined.

```
# Print the distance metrics
distance_metics <- list()
for (i in names(peaks_annotated)) {

  print(paste("The median / mean / min / max for", i, "are:"))
print(
summary(peaks_annotated[[i]]$shortestDistance))
  distance_metics[[i]] <- summary(peaks_annotated[[i]]$shortestDistance)
}
```

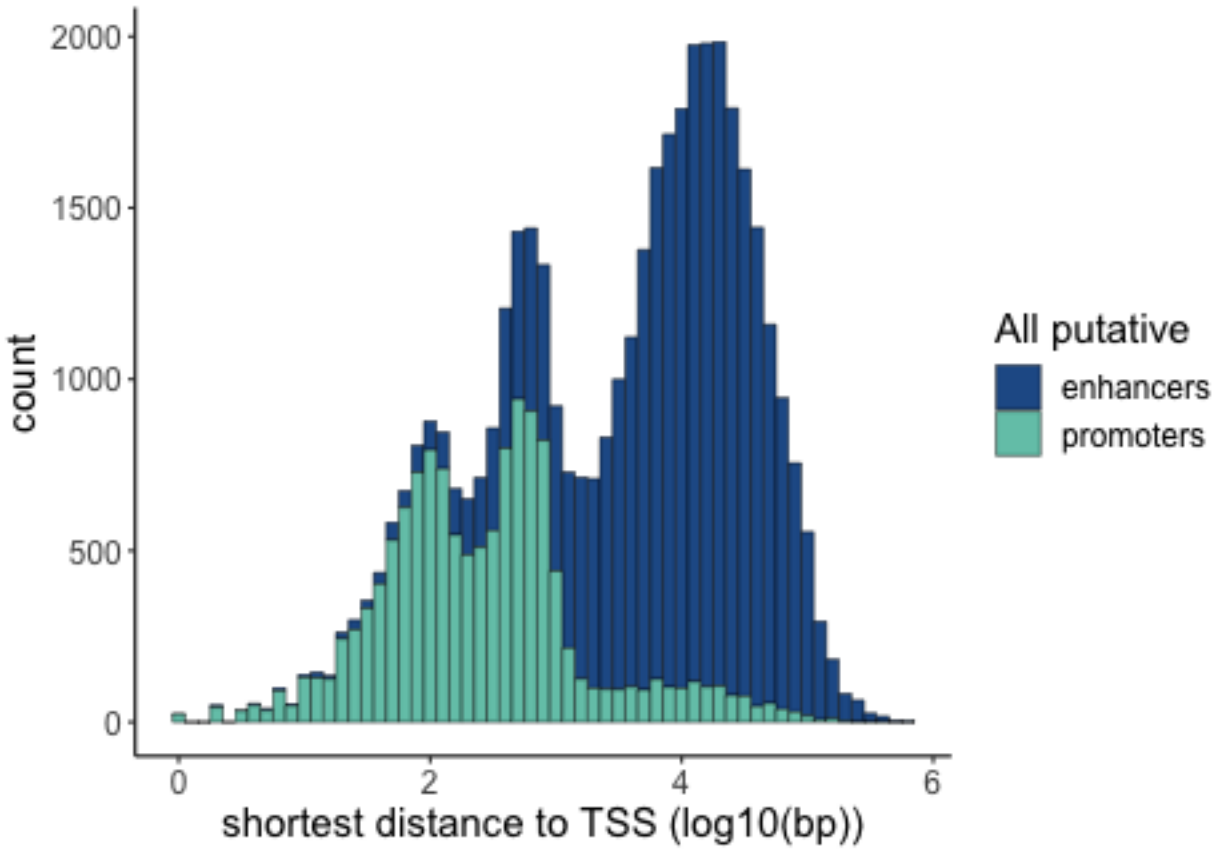
```
## [1] "The median / mean / min / max for promoters_genomewide are:"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0      74     245    2451     670  395073
## [1] "The median / mean / min / max for enhancers_genomewide are:"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    3533   11200   22274   27152  617593
## [1] "The median / mean / min / max for enhancers_allDB are:"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       5    5242   12894   25051   32282  437429
## [1] "The median / mean / min / max for promoters_allDB are:"
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##      1.0     76.0    253.5  10036.8   5304.0 256716.0
## [1] "The median / mean / min / max for enhancers_HFDup are:"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       5    4949   11838   23426   29252  281639
## [1] "The median / mean / min / max for promoters_HFDup are:"
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##      1.00     86.25    237.50  10915.61   4595.50 256716.00
## [1] "The median / mean / min / max for enhancers_HFDdown are:"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      13    6119   15100   28302   37891  437429
## [1] "The median / mean / min / max for promoters_HFDdown are:"
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##      3.0     62.5    307.0   8865.1   5304.0 126490.0
```

```
# Add a column to identify the regions
promoters_anno <- peaks_annotated$promoters_allDB %>% mutate(element = "promoters")
enhancers_anno <- peaks_annotated$enhancers_allDB %>% mutate(element = "enhancers")

promoters_genomewide_anno <- peaks_annotated$promoters_genomewide %>% mutate(element = "promoters")
enhancers_genomewide_anno <- peaks_annotated$enhancers_genomewide %>% mutate(element = "enhancers")

# Row-bind the dataframes together
combined_DBsites_anno <- rbind(promoters_anno, enhancers_anno)
combined_allSites_anno <- rbind(promoters_genomewide_anno, enhancers_genomewide_anno)
```

```
combined_DBsites_anno$shortestDistance <- combined_DBsites_anno$shortestDistance +1
combined_allSites_anno$shortestDistance <- combined_allSites_anno$shortestDistance +1

# Plot the histograms
ggplot(combined_DBsites_anno)+
  geom_histogram(aes(x=log10(shortestDistance), fill=factor(element)), binwidth=0.1, color="black", size
  theme_classic() +
  scale_fill_manual("Diff-bound\nH3K27ac at", values = c("#235b95", "#73c6b6")) +
  theme(text=element_text(size=15)) +
  xlab("shortest distance to TSS (log10(bp))")
```




```
ggplot(combined_allSites_anno)+
  geom_histogram(aes(x=log10(shortestDistance), fill=factor(element)), binwidth=0.1, color="black", size
  theme_classic() +
  scale_fill_manual("All putative", values = c("#235b95", "#73c6b6")) +
  theme(text=element_text(size=15)) +
  xlab("shortest distance to TSS (log10(bp))")
```

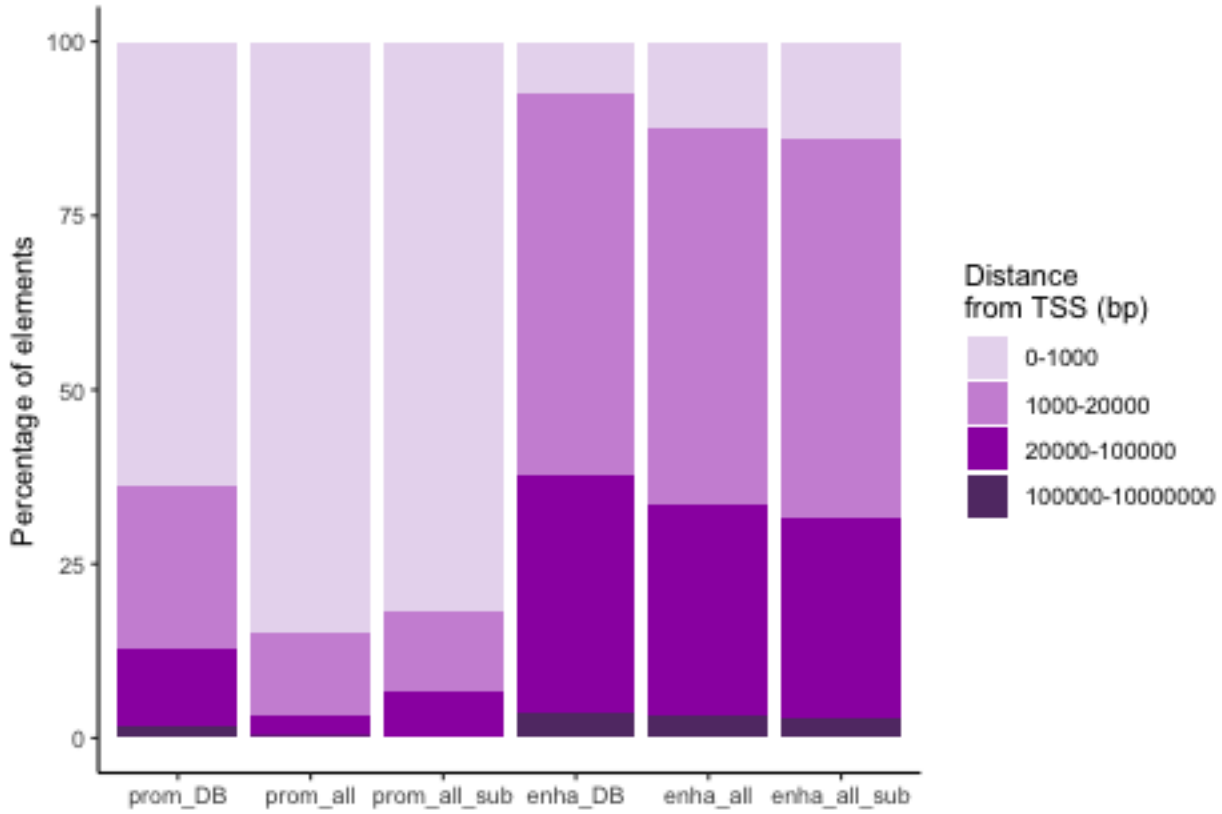


Make a stacked bar plot with the distances.

```
# set up cut-off values
breaks <- c(0,1000,20000,100000,10000000)
# specify interval/bin labels
tags <- c("[0-1000)","[1000-20000)", "[20000-100000)", "[100000-10000000)")
# bucketing values into bins
promoters_anno_bins <- cut(peaks_annotated$promoters_allDB$shortestDistance,
                  breaks=breaks,
                  include.lowest=TRUE,
                  right=FALSE,
                  labels=tags)

enhancers_anno_bins <- cut(peaks_annotated$enhancers_allDB$shortestDistance,
                  breaks=breaks,
                  include.lowest=TRUE,
                  right=FALSE,
                  labels=tags)

promoters_genomewide_anno_bins <- cut(peaks_annotated$promoters_genomewide$shortestDistance,
                  breaks=breaks,
                  include.lowest=TRUE,
                  right=FALSE,
                  labels=tags)

enhancers_genomewide_anno_bins <- cut(peaks_annotated$enhancers_genomewide$shortestDistance,
                  breaks=breaks,
```

```
                  include.lowest=TRUE,
                  right=FALSE,
                  labels=tags)

set.seed(500)
promoters_genomewide_anno_random_182 <- promoters_genomewide_anno[sample(nrow(promoters_genomewide_anno
promoters_genomewide_anno_random_182_bins <- cut(promoters_genomewide_anno_random_182$shortestDistance,
                  breaks=breaks,
                  include.lowest=TRUE,
                  right=FALSE,
                  labels=tags)

enhancers_genomewide_anno_random_1816 <- enhancers_genomewide_anno[sample(nrow(enhancers_genomewide_anno
promoters_genomewide_anno_random_1816_bins <- cut(enhancers_genomewide_anno_random_1816$shortestDistanc
                  breaks=breaks,
                  include.lowest=TRUE,
                  right=FALSE,
                  labels=tags)

# inspect bins. The occurence of each bin is counted.
summary_distance <- data.frame(prom_DB = summary(promoters_anno_bins),
           enha_DB = summary(enhancers_anno_bins),
           prom_all = summary(promoters_genomewide_anno_bins),
           prom_all_sub = summary(promoters_genomewide_anno_random_182_bins),
           enha_all = summary(enhancers_genomewide_anno_bins),
           enha_all_sub = summary(promoters_genomewide_anno_random_1816_bins))

# Getting the colsums for number of elements
summary_distance.mat <- as.matrix(summary_distance)
colsums <- colSums(summary_distance[1:6])

# .. and normalize to the colSums to get percentages
summary_distance.mat <- as.data.frame(round((
  sweep(summary_distance.mat,2,colsums, "/")*100)
  ,2))

summary_distance <- tibble::rownames_to_column(summary_distance.mat)

# long format needed for plotting
summary_distance_long <- pivot_longer(summary_distance, cols = 2:7) %>%
  group_by(name)

# Replacing the brackets
summary_distance_long$rowname <- gsub(pattern="\\[", "", summary_distance_long$rowname)
summary_distance_long$rowname <- gsub(pattern="\\)", "", summary_distance_long$rowname)

# Setting the order for plotting
order_dist <- c("0-1000", "1000-20000", "20000-100000", "100000-10000000")
order_element <- c("prom_DB", "prom_all", "prom_all_sub", "enha_DB", "enha_all", "enha_all_sub")

# Plotting the stacked bar plots
ggplot(summary_distance_long) +
  geom_col(aes(y=value, x=factor(name, levels=order_element), fill=factor(rowname, levels=order_dist)))
```

```
  scale_fill_manual("Distance\nfrom TSS (bp)", values=c("#e8daef","#ce93d8", "#9c27b0","#633974")) +
  ylab("Percentage of elements") +
  xlab("") +
  theme(text=element_text(size=15)) +
  theme_classic()
```




```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur ... 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] data.table_1.14.2    forcats_0.5.2        stringr_1.4.1
```

```
##  [4] dplyr_1.0.9          purrr_0.3.4          readr_2.1.2
##  [7] tidyr_1.2.0          tibble_3.1.8         ggplot2_3.3.6
## [10] tidyverse_1.3.2      biomaRt_2.52.0       org.Mm.eg.db_3.15.0
## [13] AnnotationDbi_1.58.0 Biobase_2.56.0       ChIPpeakAnno_3.30.1
## [16] GenomicRanges_1.48.0 GenomeInfoDb_1.32.4  IRanges_2.30.1
## [19] S4Vectors_0.34.0     BiocGenerics_0.42.0
##
## loaded via a namespace (and not attached):
##   [1] googledrive_2.0.0           colorspace_2.0-3
##   [3] rjson_0.2.21                ellipsis_0.3.2
##   [5] futile.logger_1.4.3         XVector_0.36.0
##   [7] fs_1.5.2                    rstudioapi_0.14
##   [9] farver_2.1.1                bit64_4.0.5
##  [11] lubridate_1.8.0             fansi_1.0.3
##  [13] xml2_1.3.3                  codetools_0.2-18
##  [15] splines_4.2.1               cachem_1.0.6
##  [17] knitr_1.40                  jsonlite_1.8.0
##  [19] Rsamtools_2.12.0            broom_1.0.0
##  [21] dbplyr_2.2.1                png_0.1-7
##  [23] graph_1.74.0                compiler_4.2.1
##  [25] httr_1.4.4                  backports_1.4.1
##  [27] assertthat_0.2.1            Matrix_1.4-1
##  [29] fastmap_1.1.0               lazyeval_0.2.2
##  [31] gargle_1.2.0                cli_3.3.0
##  [33] formatR_1.12                htmltools_0.5.3
##  [35] prettyunits_1.1.1           tools_4.2.1
##  [37] gtable_0.3.0                glue_1.6.2
##  [39] GenomeInfoDbData_1.2.8      rappdirs_0.3.3
##  [41] Rcpp_1.0.9                  cellranger_1.1.0
##  [43] vctrs_0.4.1                 Biostrings_2.64.1
##  [45] multtest_2.52.0             rtracklayer_1.56.1
##  [47] xfun_0.32                   rvest_1.0.3
##  [49] lifecycle_1.0.1             restfulr_0.0.15
##  [51] ensembldb_2.20.2            googlesheets4_1.0.1
##  [53] XML_3.99-0.10               InteractionSet_1.24.0
##  [55] zlibbioc_1.42.0             MASS_7.3-57
##  [57] scales_1.2.1                BSgenome_1.64.0
##  [59] hms_1.1.2                   MatrixGenerics_1.8.1
##  [61] ProtGenerics_1.28.0         parallel_4.2.1
##  [63] SummarizedExperiment_1.26.1 RBGL_1.72.0
##  [65] AnnotationFilter_1.20.0     lambda.r_1.2.4
##  [67] yaml_2.3.5                  curl_4.3.2
##  [69] memoise_2.0.1               stringi_1.7.8
##  [71] RSQLite_2.2.16              highr_0.9
##  [73] BiocIO_1.6.0                GenomicFeatures_1.48.3
##  [75] filelock_1.0.2              BiocParallel_1.30.3
##  [77] rlang_1.0.4                 pkgconfig_2.0.3
##  [79] matrixStats_0.62.0          bitops_1.0-7
##  [81] evaluate_0.16               lattice_0.20-45
##  [83] labeling_0.4.2              GenomicAlignments_1.32.1
##  [85] bit_4.0.4                   tidyselect_1.1.2
##  [87] magrittr_2.0.3              R6_2.5.1
##  [89] generics_0.1.3              DelayedArray_0.22.0
##  [91] DBI_1.1.3                   withr_2.5.0
```

```
##  [93] haven_2.5.1               pillar_1.8.1
##  [95] survival_3.3-1            KEGGREST_1.36.3
##  [97] RCurl_1.98-1.8            modelr_0.1.9
##  [99] crayon_1.5.1              futile.options_1.0.1
## [101] utf8_1.2.2                BiocFileCache_2.4.0
## [103] tzdb_0.3.0                rmarkdown_2.16
## [105] progress_1.2.2            readxl_1.4.1
## [107] grid_4.2.1                blob_1.2.3
## [109] reprex_2.0.2              digest_0.6.29
## [111] VennDiagram_1.7.3         regioneR_1.28.0
## [113] munsell_0.5.0
```